

# ÖVNING I: INTRODUKTION TILL ACTIONSCRIPT 3.0

Tidigare har vi gjort all programmering av ActionScript 3.0 i tidslinjen i Flash. Från och med nu kommer vi dock att ha minst två olika filer för kommande övningar, minst en AS-fil och en FLA-fil.

- **AS** – Denna fil innehåller all din AS3.0 kod. AS-filer är textfiler som har filändelsen .as. Tidigare har vi skrivit all AS-kod direkt i olika keyframes i FLA-filen och endast använt oss av en fil. Fördelen med att använda en separat fil för AS är att du slipper hålla reda på olika lager och keyframes där AS ska gälla och du får ett mer strukturerat och lättöverskådligt sätt att programmera på.
- **FLA** – detta är en vanlig FLA-fil och har filändelsen .fla. FLA-filen innehåller alla dina visuella objekt som ditt program kommer att innehålla. Det är AS-filen som kommer att stå för allt ”tänkande” men den är beroende av FLA-filen för att kunna tolka och köra koden. FLA-filen kan också innehålla inbäddade objekt såsom animeringar, ljud, video och grafik.

## ÖVNING I AV 2

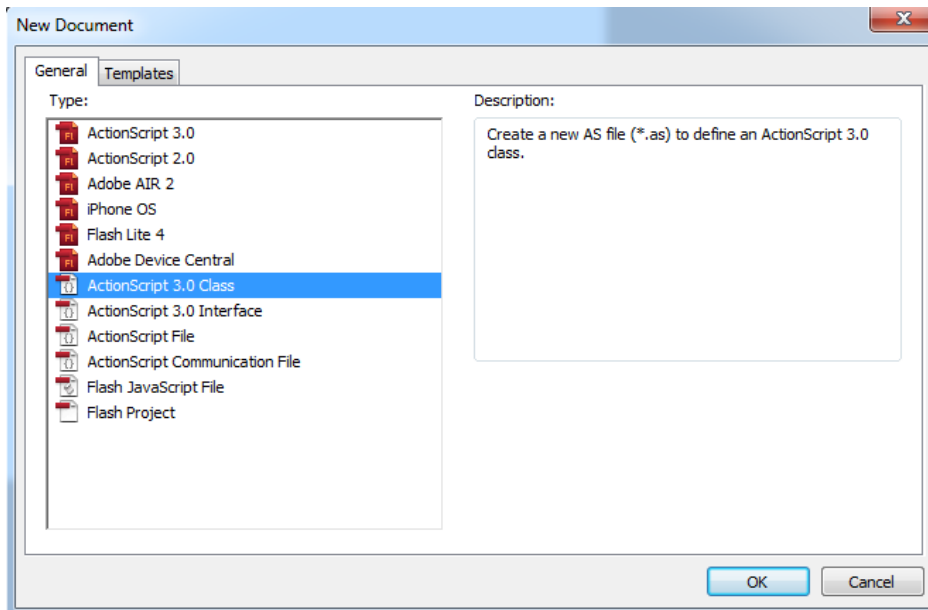
---

Du ska nu skapa ett litet program som skriver ut ”Hello World”.

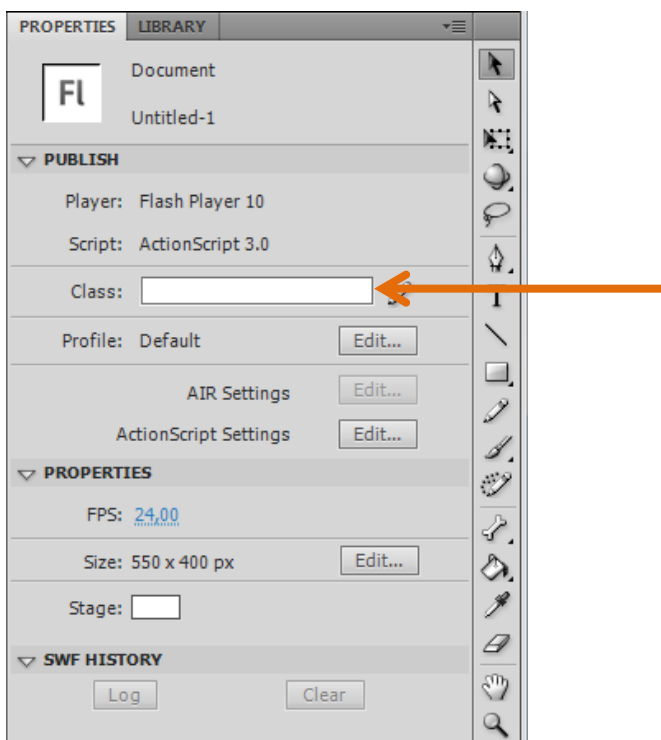
1. Starta Adobe Flash. Välj att skapa en ny ActionScript 3.0 fil (precis som vanligt). En .FLA-fil skapas. Välj File → Save as... och döp filen till HelloWorld.



2. Gå till File → New... och välj ActionScript 3.0 Class. I rutan som visas, ange namnet Main. En .AS-fil skapas.



3. Välj File → Save. Om allt stämmer ska det redan stå Main som filnamn. Om det inte gör det, döp filen till Main. Se till att **spara filen i samma mapp** som .FLA-filen!
4. Återgå till den första filen du skapade (.FLA-filen).
5. Ange Main som class i properties.



6. Spara filen (Ctrl + S).
7. Gå till Main.as. Den ser ut på följande sätt just nu:

```

1 package {
2
3     public class Main {
4
5         public function Main() {
6             // constructor code
7         }
8
9     }
10
11 }
12

```

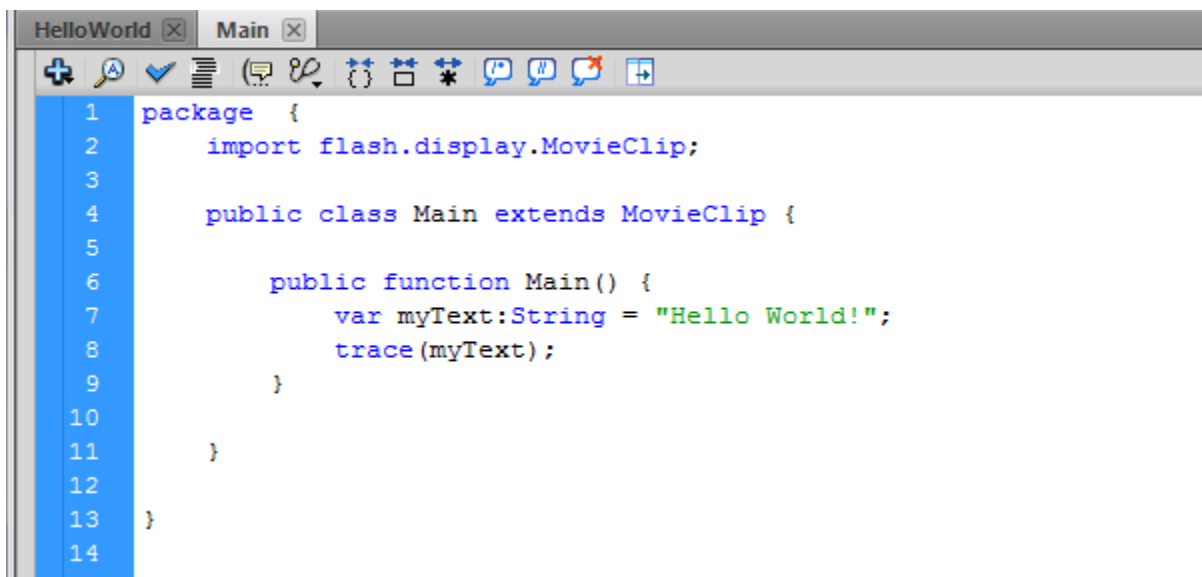
8. Som du ser finns redan ett färdigt package, och klassen är döpt till Main. Det finns t.o.m. en constructor färdig. Lägg till följande kod i constructor-metoden:

```

var myText:String = "Hello World!";
trace(myText);

```

9. Spara. Om du nu skulle testa din applikation (Ctrl + Enter) skulle du få ett fel och ingenting skulle fungera. Det är nämligen en grej till vi måste göra. För att vi ska kunna använda trace() måste vi använda en klass som kan hantera den funktionen, t.ex. MovieClip. Därför måste vi ange att vår klass "extends" MovieClip. Lägg till det som behövs för att din klass ska se ut på följande sätt:



```

HelloWorld x Main x
+ [A] [✓] [≡] [⌂] [⌘] [⌘] [⌘] [⌘] [⌘] [⌘] [⌘] [⌘]
1 package {
2     import flash.display.MovieClip;
3
4     public class Main extends MovieClip {
5
6         public function Main() {
7             var myText:String = "Hello World!";
8             trace(myText);
9         }
10
11     }
12
13 }
14

```

10. Spara (Ctrl + S) och testa nu din applikation genom att trycka Ctrl + Enter (notera att du måste spara filen för att ändringarna ska visas). Om du tittar i Output-rutan ser du att det nu står "Hello World!" där.

*Trace() är en funktion som är mycket bra att ha! Du kan t.ex. skriva ut variabler när du felsöker för att se att allting stämmer. Trace() är nog din allra bästa vän när du programmerar AS 3.0.*

11. Vi ska nu träna på att deklarera variabler och använda operatörer. Ta bort variabeln myText. Skapa följande variabler:
- En string med värdet "Hello".
  - En string med värdet "World".
  - En int med värdet 2.

- d. En number utan värde.
  - e. En boolean med värdet true.
12. Tänk på att namnen du ger variablerna ska följa naming conventions.
  13. Genom att använda operatorn +, gör en trace() som skriver ut "Hello World 2" med hjälp av dina nya variabler (tips: det går att använda + även för textsträngar).
  14. Öka värdet på variabel c genom att göra +1.
  15. Öka värdet på variabel c utan att skriva +1 (kommer du ihåg hur du ska skriva då?).
  16. Ge variabel d samma värde som variabel c (genom att använda =).
  17. Subtrahera 5.5 från variabel d.
  18. Gör en trace() som skriver ut värdet på c, d och e.

Du kan se min lösning på nästa sida.

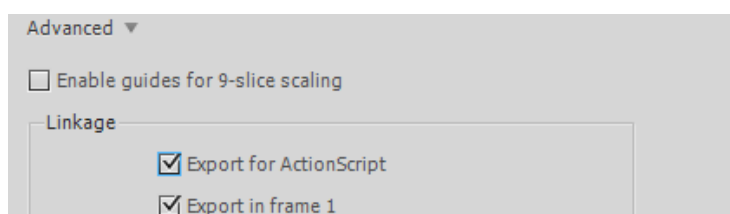
```
1 package {
2     import flash.display.MovieClip;
3
4     public class Main extends MovieClip {
5
6         public function Main() {
7             //Declares variables
8             var hello:String = "Hello";
9             var world:String = "World";
10            var two:int = 2;
11            var myNumber:Number;
12            var myBool:Boolean = true;
13
14            trace(hello + " " + world + " " + two); //Traces "Hello World 2"
15
16            two = two + 1; //Adds 1 to two
17            two++; //Increments two by one
18            myNumber = two; //Sets myNumber equal to two
19            myNumber = myNumber - 5.5; //Subtracts 5.5 from myNumber
20
21            //Traces values of two, myNumber and myBool
22            trace("two: " + two + " myNumber: " + myNumber + " myBool: " + myBool);
23        }
24    }
25 }
26
27 }
28 }
```

## ÖVNING 2 AV 2 – FÖRBEREDELSE INFÖR NÄSTA ÖVNING

---

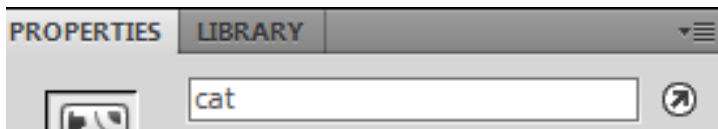
I denna övning ska du förbereda en .FLA-fil inför nästa övning, samtidigt som du repeterar funktionerna i Flash.

1. Skapa en ny mapp.
2. Skapa en ny .FLA-fil. Döp den till InteraktivBok och spara den i den nya mappen.
3. Skapa en ny .AS-fil. Döp den till Main (hade du använt samma mapp som i förra övningen hade du inte kunnat döpa den till Main...).
4. I InteraktivBok.fla anger du Main som class (i Properties).
5. Gå till Insert → New Symbol... Ange namnet HillPage, välj MovieClip. Klicka på "Advanced" och kryssa i rutan "Export for ActionScript". Mer om vad detta innebär på nästa övning.



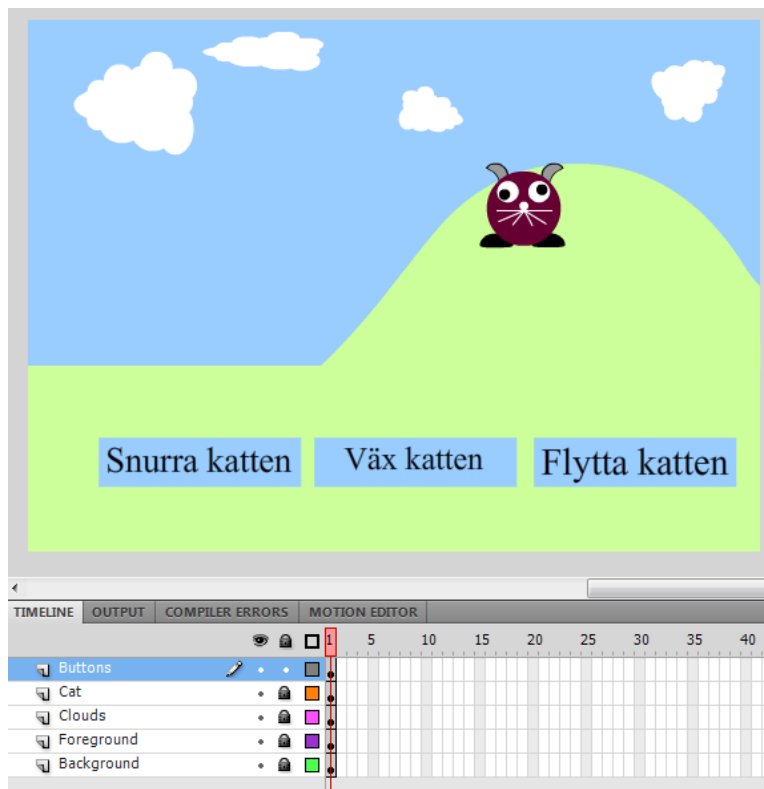
6. Klicka på OK och även OK på varningen som visas.
7. Om allt stämmer är du nu "inne i" din nya symbol. Rita följande i ditt nya Movie Clip (det är alltså i Movie Clipet du ska lägga till detta, inte på scenen):

- a. En ruta som fungerar som bakgrund (blå) (tänk på att lägga den på ett eget lager och lås lagret).
  - b. En förgrund i form av en grön kulle.
  - c. Ett moln. Gör om molnet till en graphic och dra ut några fler i Movie Clipet. Ändra storlek och rotera på dem så att de inte ser likadana ut.
8. Lägg lite tid på att göra det fint. Kom ihåg att skapa olika lager och ge lagren logiska namn. Tänk också på att hålla ordning på ditt bibliotek.
  9. Gå in på Insert → New Symbol. Ange namnet Cat (var noga med stor bokstav) och typ Movie Clip. Klicka på "Advanced" och kryssa i rutan "Export for ActionScript". Klicka på OK och även OK på varningen som visas.
  10. Om allt stämmer är du nu "inne i" din nya symbol. Rita här en liten figur som föreställer en katt.
  11. Dra ut din katt i HillPage när du är färdig.
  12. Ge katten instansnamnet cat (med små bokstäver).



13. Gå in på Insert → New Symbol. Ange namnet DownButton, type Button och kryssa även här för "Export for ActionScript". Klicka på OK (två gånger).
14. Rita grafiken till din knapp så att det är någon förändring för varje läge. Skriv någon text, t.ex. "Flytta katten".
15. Dra ut knappen till ditt Movie Clip (inte på scenen) och ge den instansnamnet downButton.
16. Skapa två knappar till och dra ut dem på scenen:
  - a. GrowButton (instansnamn: growButton) med texten "Väx katten".
  - b. RotateButton (instansnamn: rotateButton) med texten "Snurra katten".

Min HillPage ser ut på följande sätt (du får självklart göra precis hur du vill):



Det du nu har gjort kommer du att använda i nästa övning för att lägga på kod och få katten att röra sig. Se därför till att spara filen någonstans där du kommer åt den nästa gång (t.ex. ditt studentkonto eller ett USB-minne).